Configuration Manual

MSc Research Project
Data Analytics

Rajbharath Jothimani
Student ID: x21133000

School of Computing
National College of Ireland

Supervisor: Cristina Hava Muntean

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rajbharath Jothimani
Student ID:	x21133000
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Cristina Hava Muntean
Submission Due Date:	01/02/2023
Project Title:	Configuration Manual
Word Count:	1340
Page Count:	8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Rajbharath Jothimani
Date:	31st January 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

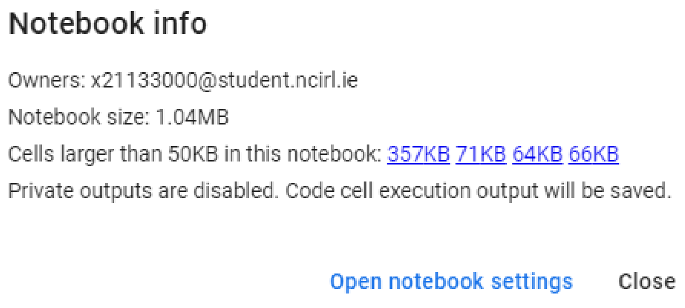
Rajbharath Jothimani
x21133000

1 Introduction

The purpose of this document is to explain the setup and implementation created for the purpose of executing this Research. It also describes more on the platform specification and tools used in this study. Moreover, it also explains much details on the python libraries used in this research for the purpose of performing the modelling techniques and also a little on the model building techniques used.

2 Notebook Configuration used in Google Colab

This research is done in the google colab environment. Hence, it does not require the specification of local host to run the code. It can be run directly from the google colab environment as seen in Figure 1 for the purpose of this research.



Notebook info

Owners: x21133000@student.ncirl.ie

Notebook size: 1.04MB

Cells larger than 50KB in this notebook: 357KB 71KB 64KB 66KB

Private outputs are disabled. Code cell execution output will be saved.

Figure 1: Notebook Configuration used in Google Colab

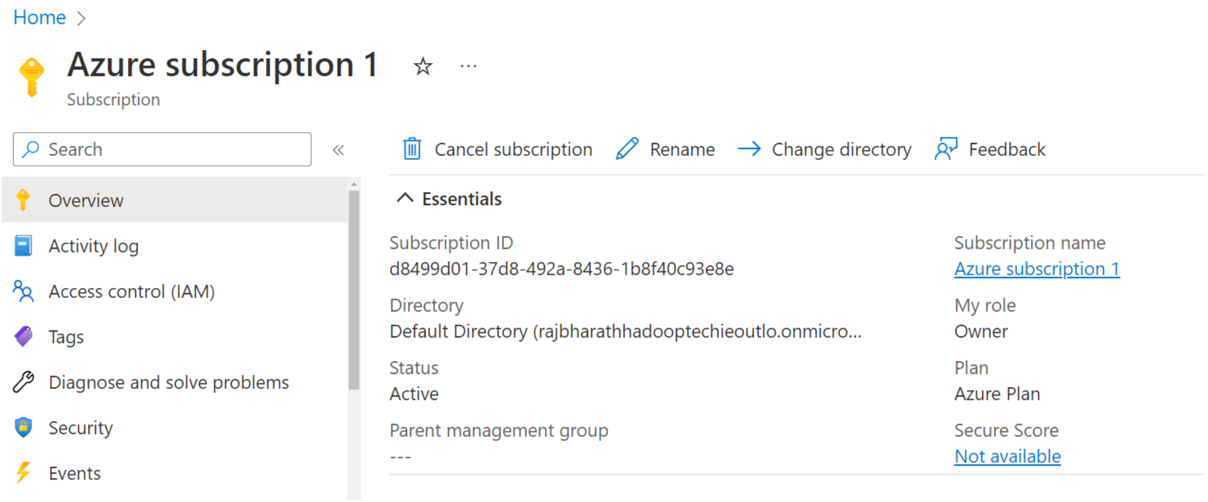
3 Collection of Dataset used in this research

The dataset used in this research is obtained from UCI machine learning repository [1]. It has a total of 18 columns and 12330 rows.

4 Microsoft Azure Cloud Storage

In-order to work with Azure cloud, a proper azure subscription is required as shown in Figure 2 below

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/00468/



Azure subscription 1

Subscription

Subscription ID
d8499d01-37d8-492a-8436-1b8f40c93e8e

Subscription name
Azure subscription 1

Directory
Default Directory (rajbharathhadooptechieoutlo.onmicro...

My role
Owner

Status
Active

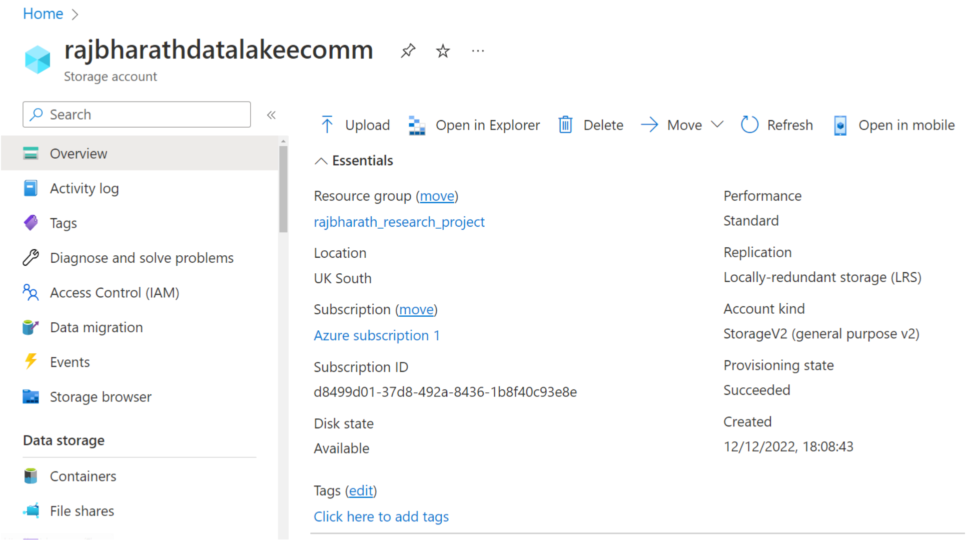
Plan
Azure Plan

Parent management group

Secure Score
Not available

Figure 2: Azure subscription in the azure cloud.

The data used in this research is stored in Azure data lake storage for the purpose of quicker fetch and security reasons. The details of the Azure data lake storage account are shown in Figure 3 mentioned below. Azure data lake storage in the name of rajbharathdatalakeecomm is created and attached to the resource group rajbharath_research_project. Hierarchial namespace and Blob public access is enabled as seen in Figure 3.



rajbharathdatalakeecomm

Storage account

Resource group (move)
rajbharath_research_project

Performance
Standard

Location
UK South

Replication
Locally-redundant storage (LRS)

Subscription (move)
Azure subscription 1

Account kind
StorageV2 (general purpose v2)

Subscription ID
d8499d01-37d8-492a-8436-1b8f40c93e8e

Provisioning state
Succeeded

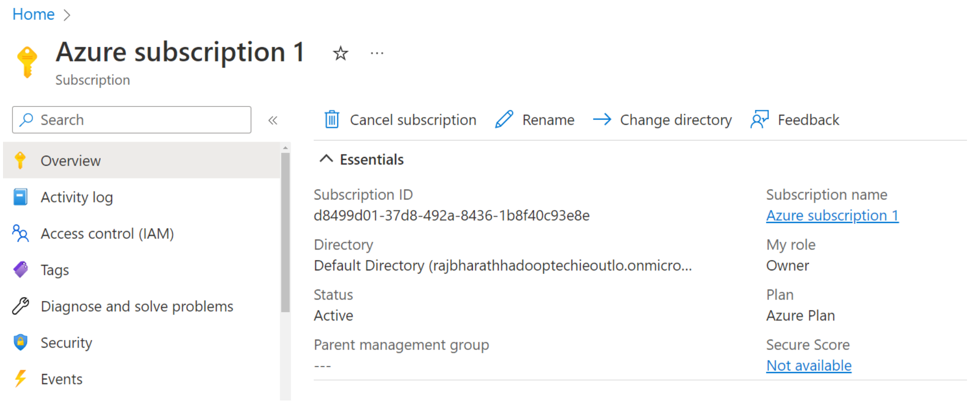
Disk state
Available

Created
12/12/2022, 18:08:43

Tags (edit)
Click here to add tags

Figure 3: Implementation of Azure Data Lake Storage in the cloud

Later a container is created inside the data lake storage and the input CSV file is uploaded into the container as seen in Figure 4.



Subscription ID
d8499d01-37d8-492a-8436-1b8f40c93e8e

Subscription name
Azure subscription 1

Directory
Default Directory (rajbharathhadooptechieoutlo.onmicro...

My role
Owner

Status
Active

Plan
Azure Plan

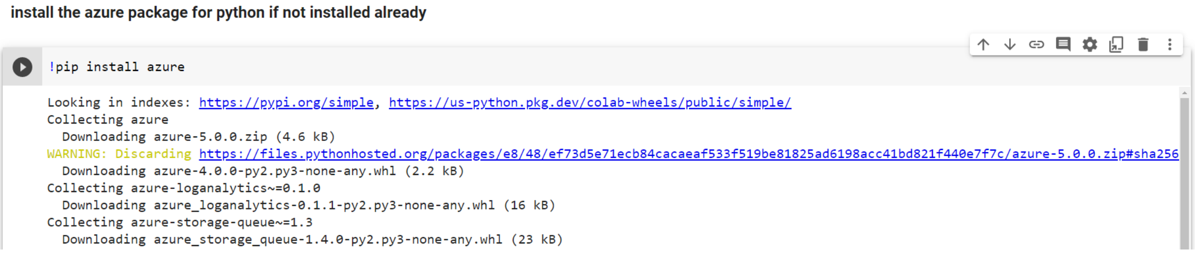
Parent management group

Secure Score
Not available

Figure 4: CSV file stored as blob in Azure Data Lake Storage in the cloud

5 Connecting and Fetching from ADLS

Data has to be fetched from Azure Data Lake storage and Loaded into Google Colab platform. Hence, Azure python package is installed in the Google colab as seen below in figure 5

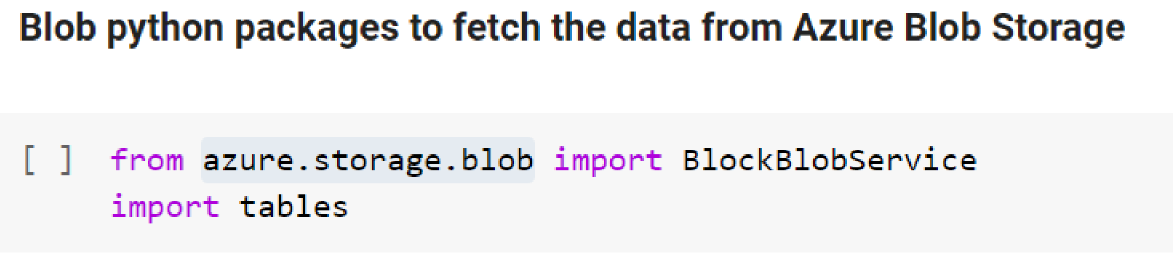

install the azure package for python if not installed already

```
!pip install azure
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting azure
  Downloading azure-5.0.0.zip (4.6 kB)
WARNING: Discarding https://files.pythonhosted.org/packages/e8/48/ef73d5e71ecb84cacaeaf533f519be81825ad6198acc41bd821f440e7f7c/azure-5.0.0.zip#sha256
  Downloading azure-4.0.0-py2.py3-none-any.whl (2.2 kB)
Collecting azure-loganalytics~=0.1.0
  Downloading azure_loganalytics-0.1.1-py2.py3-none-any.whl (16 kB)
Collecting azure-storage-queue~=1.3
  Downloading azure_storage_queue-1.4.0-py2.py3-none-any.whl (23 kB)
```

Figure 5: Installing Azure package in the google cloud

BlockBlobService method from azure.storage.blob library is required to extract blob from the azure data lake storage. Hence , it is imported into the python program as seen in Figure 6. The blob fetched is directly converted into a pandas dataframe.


Blob python packages to fetch the data from Azure Blob Storage

```
[ ]  from azure.storage.blob import BlockBlobService
     import tables
```

Figure 6: BlockBlobService to extract blob from the azure data lake storage

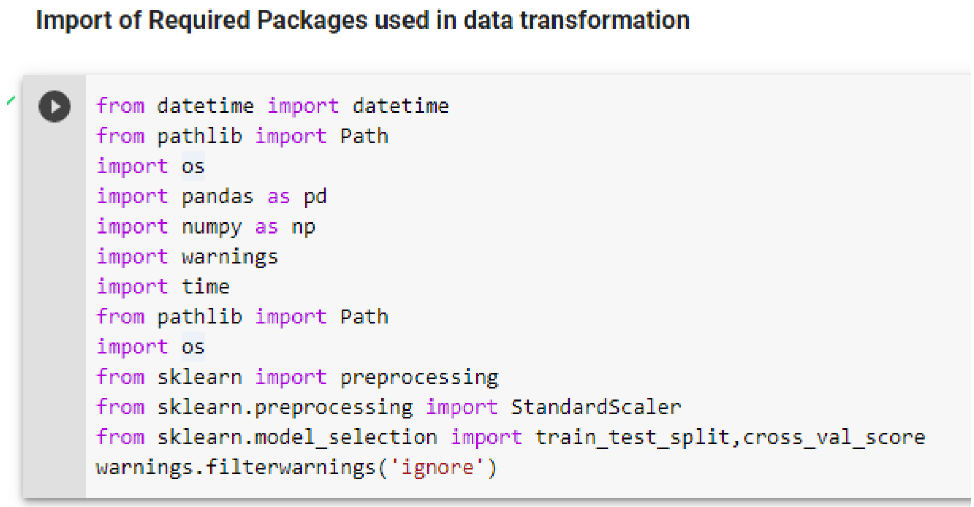
6 Import of all the required python packages

Various python packages are required in performing the data pre-processing,cleaning, transformation, modelling, visualization and evaluation phases of the research. All these required python packages are imported into the program as seen in the images mentioned in the subsections below.

In addition to this, various packages might not be installed in the environment already. Hence, these packages have to be installed prior to using these packages in python.

6.1 Importing all the Required Packages used in data transformation

Data transformation needs to be done to conduct the research as per the requirements. Hence, various python packages are required as seen in Figure 7



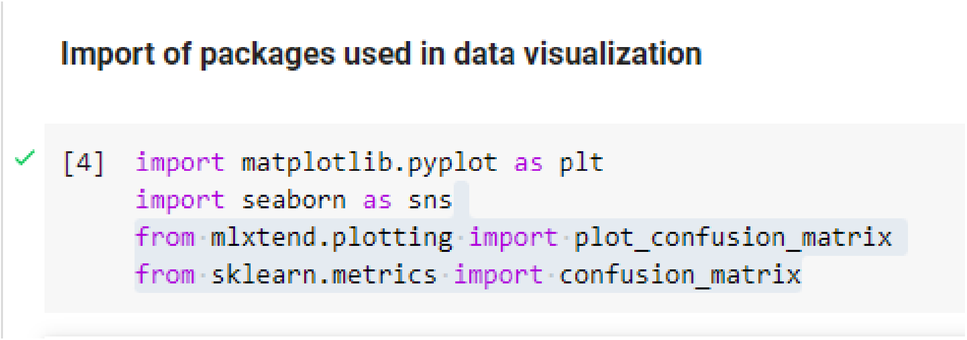
Import of Required Packages used in data transformation

```
from datetime import datetime
from pathlib import Path
import os
import pandas as pd
import numpy as np
import warnings
import time
from pathlib import Path
import os
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split,cross_val_score
warnings.filterwarnings('ignore')
```

Figure 7: Packages used in data transformation

6.2 Import of Required Packages used in data visualization

Data visualizations are essential to create plots such as correlation plots, box plots, frequency plots and confusion matrix. All the required python packages used in this research for visualization are shown in the figure 8. Visualization is very important since it results in the creation of actions plans leading to better business decisions Various python packages are dedicated for visualization needs in-order to perform better analytics on data to obtain a better results. Matplotlib and Seaborn are few of the commonly used packages in python that can aid in data visualizations in-order to obtain better decisions. In this research both the packages are used in-order to create wide range of visualizations for the purpose of predicting the arrival of festival using user session data of e-commerce



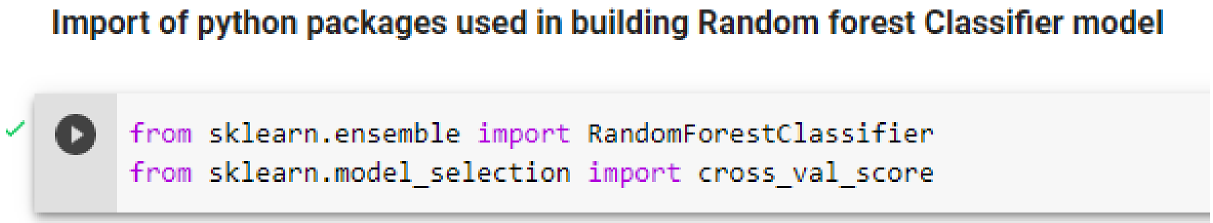
Import of packages used in data visualization

```
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
```

Figure 8: Packages used in data visualization

6.3 Import of Required Packages used in building Random forest Classifier model

Random forest Classifier is one of the model selected to work on this prediction problem because of its accuracy in performing the prediction since it involves usage of multitude of decision trees for the purpose of prediction. Also, various hyper-parameter tuning is done in Random forest classifier using 7 fold cross validation. Hence, cross validation score is also essential in choosing the hyper parameter . Hence, the required packages for cross validation should also be implemented as seen in Figure 9



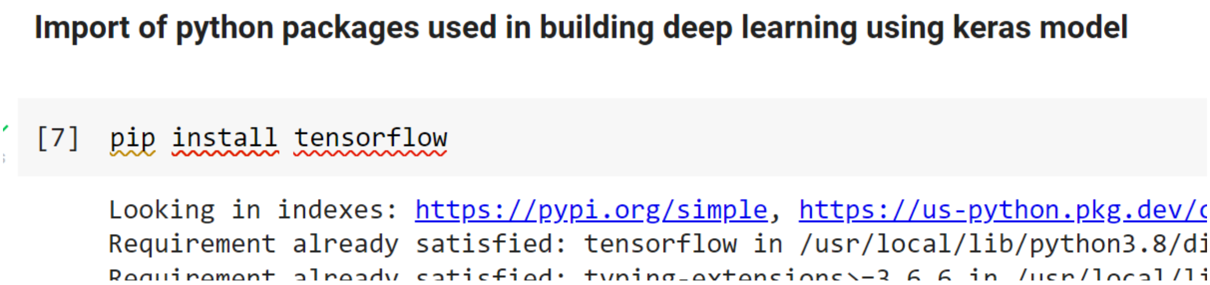
Import of python packages used in building Random forest Classifier model

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
```

Figure 9: Packages used in building Random forest Classifier model

6.4 Import of Required Packages used in building deep learning using keras

Deep learning using keras is also chosen for this prediction problem. Usage of deep learning model with keras requires installation of tensor-flow package. Hence, the tensorflow package is first installed before importing all the required python packages of keras to use in the modelling for the purpose of prediction as seen in figure 10. Sequential modelling is preferred to be used in this model. Dense layers using Sigmoid and Relu activation functions are used in this modelling. Adam optimizer with binary_crossentropy loss function is used . The python packages used in this research are displayed in the figure 11. Keras is one of the open source library developed for the creation and evaluation of machine learning models. It is one of the essential component of tensorflow framework which is specifically created of developing neural networks at easy with less code environments. This require Python 2 or Python 3 to be installed in the machine. In this method, Sequential model is used, where the layers are added one after the other in a sequential order until a better prediction is obtained. Finally, a fully connected network with three different layers is used in this modelling process.

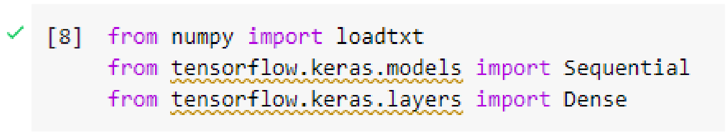


Import of python packages used in building deep learning using keras model

```
[7] pip install tensorflow

    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/c
    Requirement already satisfied: tensorflow in /usr/local/lib/python3.8/di
    Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/li
```

Figure 10: Installation of Tensor flow library in python

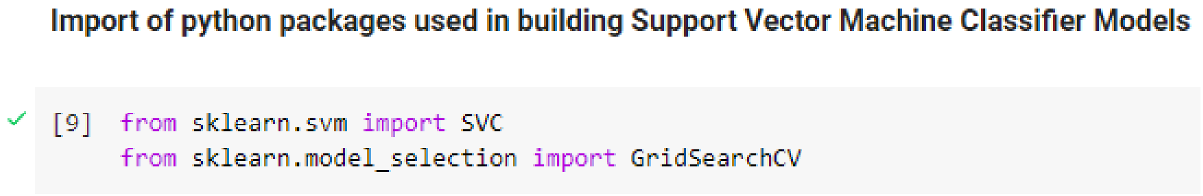


```
[8] from numpy import loadtxt
    from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense
```

Figure 11: Packages used in building deep learning using keras

6.5 Import of Required Packages used in building support vector machine classifier

All the required python libraries used by the support vector machine classifier are shown in the figure 12. Suport vector machine classifier is also one of the preferred classification machine learning model for performing this prediction.



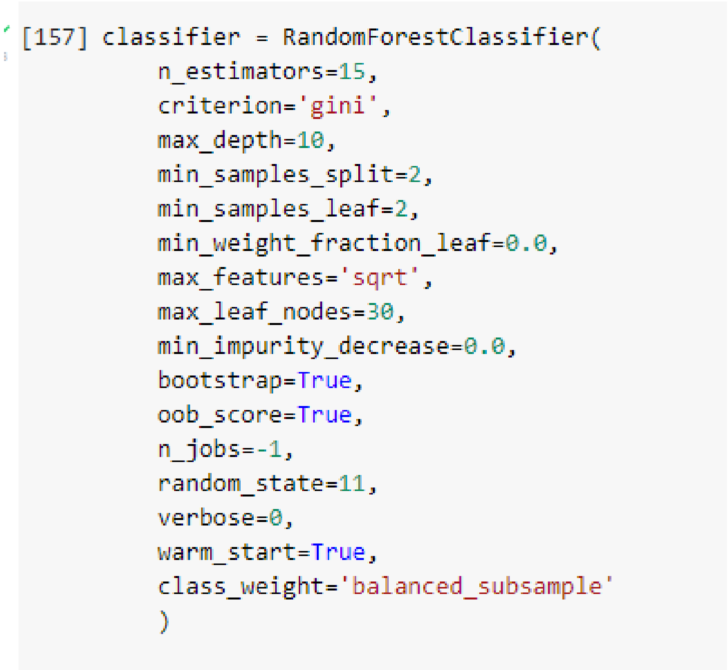
Import of python packages used in building Support Vector Machine Classifier Models

```
[9] from sklearn.svm import SVC
    from sklearn.model_selection import GridSearchCV
```

Figure 12: Packages used in building support vector machine classifier

7 Hyper-parameters used in Random Forest Classifier

The below image in Figure 13 shows all the parameter used in hyper-parameter tuning in order to improve the accuracy of the prediction. These parameters should be to set in order to obtain better prediction accuracy in Random Forest Classifier as seen in Fig mentioned below.

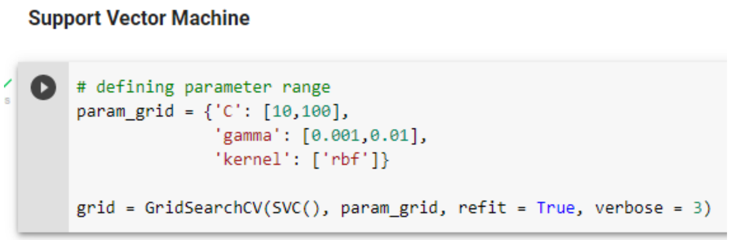


```
[157] classifier = RandomForestClassifier(
              n_estimators=15,
              criterion='gini',
              max_depth=10,
              min_samples_split=2,
              min_samples_leaf=2,
              min_weight_fraction_leaf=0.0,
              max_features='sqrt',
              max_leaf_nodes=30,
              min_impurity_decrease=0.0,
              bootstrap=True,
              oob_score=True,
              n_jobs=-1,
              random_state=11,
              verbose=0,
              warm_start=True,
              class_weight='balanced_subsample'
              )
```

Figure 13: Hyper-parameters used in Random Forest Classifier

8 Hyper-parameters used in Support Vector Machine Classifier

The below image in Figure 14 shows all the parameter used in hyper-parameter tuning in order to improve the accuracy of the prediction in Support Vector Machine Classifier. These parameters should be to set in-order to obtain better prediction accuracy in Support Vector Machine Classifier as seen in Fig mentioned below.



Support Vector Machine

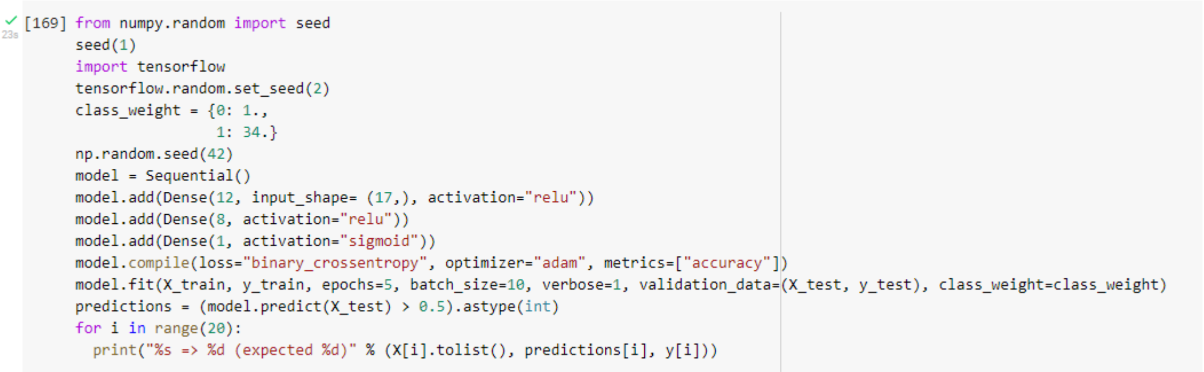
```
# defining parameter range
param_grid = {'C': [10,100],
              'gamma': [0.001,0.01],
              'kernel': ['rbf']}

grid = GridSearchCV(SVC(), param_grid, refit = True, verbose = 3)
```

Figure 14: Hyper-parameters used in Support Vector Machine Classifier

9 Optimization of deep learning using Keras

The below images shows all the layers used in deep learning with Keras in order to improve the accuracy of the prediction in deep learning with Keras involved in classification problem. These parameters should be set in-order to obtain better prediction accuracy in deep learning using Keras as seen in Figure 15 mentioned below. Adam optimizer and dense layers are used in the modelling process to improve the performance of the model to obtain better prediction results.

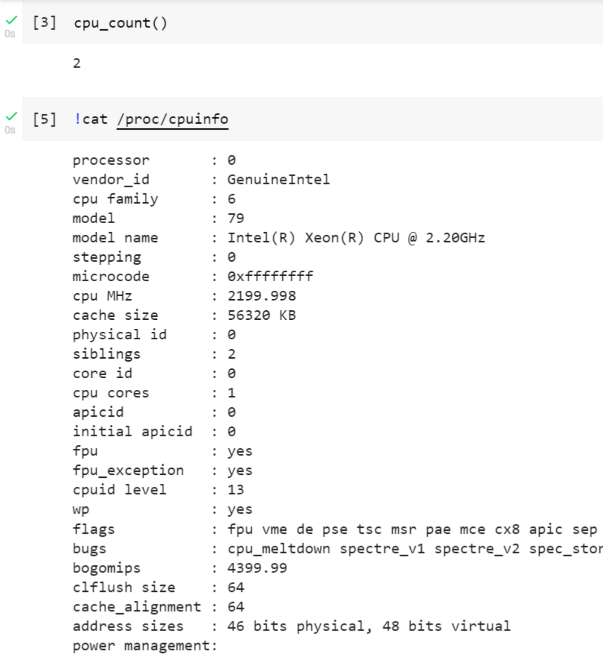


```
[169] from numpy.random import seed
      seed(1)
      import tensorflow
      tensorflow.random.set_seed(2)
      class_weight = {0: 1.,
                      1: 34.}
      np.random.seed(42)
      model = Sequential()
      model.add(Dense(12, input_shape= (17,), activation="relu"))
      model.add(Dense(8, activation="relu"))
      model.add(Dense(1, activation="sigmoid"))
      model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])
      model.fit(X_train, y_train, epochs=5, batch_size=10, verbose=1, validation_data=(X_test, y_test), class_weight=class_weight)
      predictions = (model.predict(X_test) > 0.5).astype(int)
      for i in range(20):
        print("%s => %d (expected %d)" % (X[i].tolist(), predictions[i], y[i]))
```

Figure 15: Hyper-parameters used in deep learning with Keras

10 CPU Configuration setup used by Google Colab

CPU configuration set used by Google colab is dispalyed in Figure 16 as shown below.



```
[3]  cpu_count()

     2

[5]  !cat /proc/cpuinfo

     processor       : 0
     vendor_id       : GenuineIntel
     cpu family      : 6
     model           : 79
     model name      : Intel(R) Xeon(R) CPU @ 2.20GHz
     stepping        : 0
     microcode       : 0xffffffff
     cpu MHz         : 2199.998
     cache size      : 56320 KB
     physical id     : 0
     siblings        : 2
     core id         : 0
     cpu cores       : 1
     apicid          : 0
     initial apicid  : 0
     fpu             : yes
     fpu_exception   : yes
     cpuid level     : 13
     wp              : yes
     flags           : fpu vme de pse tsc msr pae mce cx8 apic sep
     bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_stor
     bogomips        : 4399.99
     clflush size    : 64
     cache_alignment : 64
     address sizes   : 46 bits physical, 48 bits virtual
     power management:
```

Figure 16: CPU Configuration setup used by Google Colab